Algorithmic Guarantees for Inverse Imaging with Untrained Neural
Network Priors

Gauri Bhagawantrao Jagatap
Chinmay Hegde
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/creativecomponents

Part of the Signal Processing Commons

Recommended Citation
https://lib.dr.iastate.edu/creativecomponents/400

This Creative Component is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Creative Components by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Algorithmic Guarantees for Inverse Imaging with Untrained Neural Network Priors

by

Gauri Bhagawantrao Jagatap

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering

Program of Study Committee:
Dr. Chinmay Hegde, Major Professor

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation/thesis. The Graduate College will ensure this dissertation/thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2019

Copyright © Gauri Bhagawantrao Jagatap, 2019. All rights reserved.
Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>iv</td>
</tr>
<tr>
<td>Chapter 1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Our contributions</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Prior work</td>
<td>4</td>
</tr>
<tr>
<td>Chapter 2. NOTATION</td>
<td>6</td>
</tr>
<tr>
<td>Chapter 3. PROBLEM SETUP</td>
<td>7</td>
</tr>
<tr>
<td>3.1 Deep network priors</td>
<td>7</td>
</tr>
<tr>
<td>3.2 Observation models and assumptions</td>
<td>8</td>
</tr>
<tr>
<td>Chapter 4. LINEAR COMPRESSIVE SENSING WITH DEEP UNTRAINED NETWORK PRIOR</td>
<td>11</td>
</tr>
<tr>
<td>Chapter 5. COMPRESSIVE PHASE RETRIEVAL WITH DEEP UNTRAINED NETWORK PRIOR</td>
<td>13</td>
</tr>
<tr>
<td>5.1 PROJECTION TO DEEP UNTRAINED NETWORK PRIOR</td>
<td>16</td>
</tr>
<tr>
<td>Chapter 6. EXPERIMENT RESULTS</td>
<td>17</td>
</tr>
<tr>
<td>6.1 Compressive sensing</td>
<td>18</td>
</tr>
<tr>
<td>6.2 Compressive phase retrieval</td>
<td>19</td>
</tr>
<tr>
<td>Chapter 7. DISCUSSION ON SAMPLE COMPLEXITY</td>
<td>21</td>
</tr>
<tr>
<td>Chapter 8. PROOFS OF MAIN LEMMAS AND THEOREMS</td>
<td>22</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>30</td>
</tr>
<tr>
<td>A. SUPPLEMENTARY LEMMAS</td>
<td>33</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Chinmay Hegde for his guidance, patience and support throughout this research and the writing of this thesis. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank Dr. Namrata Vaswani for her mentorship on several projects.
ABSTRACT

Deep neural networks as image priors have been recently introduced for problems such as denoising, super-resolution and inpainting with promising performance gains over hand-crafted image priors such as sparsity and low-rank. Unlike learned generative priors they do not require any training over large datasets. However, few theoretical guarantees exist in the scope of using untrained network priors for inverse imaging problems. We explore new applications and theory for untrained neural network priors. Specifically, we consider the problem of solving linear inverse problems, such as compressive sensing, as well as non-linear problems, such as compressive phase retrieval. We model images to lie in the range of an untrained deep generative network with a fixed seed. We further present a projected gradient descent scheme that can be used for both compressive sensing and phase retrieval and provide rigorous theoretical guarantees for its convergence. We also show both theoretically as well as empirically that with deep network priors, one can achieve better compression rates for the same image quality as compared to when hand crafted priors are used.\footnote{This work has been accepted for publication and presentation at the Thirty-third Conference on Neural Information Processing Systems (NeurIPS), 2019 : G. Jagatap and C. Hegde, “Algorithmic Guarantees for Inverse Imaging with Untrained Network Priors.”}
Chapter 1. INTRODUCTION

1.1 Motivation

Deep neural networks have led to unprecedented success in solving several problems, specifically in the domain of inverse imaging. Image denoising [Vincent et al. (2010)], super-resolution [Dong et al. (2016)], inpainting and compressed sensing Chang et al. (2017), and phase retrieval [Metzler et al. (2018)] are among the many imaging applications that have benefited from the usage of deep convolutional networks (CNNs) trained with thousands of images.

Apart from supervised learning, deep CNN models have also been used in unsupervised setups, such as Generative Adversarial Networks (GANs). Here, image priors based on a generative model [Goodfellow et al. (2014)] are learned from training data. In this context, neural networks emulate the probability distribution of the data inputs. GANs have been used to model signal prior by learning the distribution of training data. Such learned priors have replaced hand-crafted priors with high success rates [Chang et al. (2017); Bora et al. (2017); Hand et al. (2018); Y. Wu (2019)]. However, the main challenge with these approaches is the requirement of massive amounts of training data. For instance, super-resolution CNN [Dong et al. (2016)] uses ImageNet which contains millions of images. Moreover, convergence guarantees for training such networks are limited [Hand et al. (2018)].

In contrast, there has been recent interest in using untrained neural networks as an image prior. Deep Image Prior [Ulyanov et al. (2018)] and variants such as Deep Decoder [Heckel and Hand (2018)] are capable of solving linear inverse imaging problems with no training data whatsoever, while merely imposing an auto-encoder [Ulyanov et al. (2018)] and decoder [Heckel and Hand (2018)] architecture as a structural prior. For denoising, inpainting and super-resolution, deep image priors have shown superior reconstruction performance as compared to conventional methodologies such as basis pursuit denoising (BPDN) [Chen et al. (2001)], BM3D [Dabov et al. (2006)] as well as
convolutional sparse coding [Papyan et al. (2017)]. Similar empirical results have been claimed very recently in the context of time-series data for audio applications [S. Ravula (2019); M. Michelashvili (2019)]. The theme in all of these approaches is the same: to design a prior that exploits local image correlation, instead of global statistics, and find a good low-dimensional neural representation of natural images. However, most of these works have very limited [Sulam et al. (2018); Heckel and Hand (2018)] or no theoretical guarantees.

Neural networks priors for compressive imaging has only recently been explored. In the context of compressive sensing (CS), [Van Veen et al. (2018)] uses Deep Image Prior along with learned regularization for reconstructing images from compressive measurements [Donoho (2006)]. However, the model described still relies on training data for learning appropriate regularization parameters. For the problem of compressive sensing, priors such as sparsity [Needell and Tropp (2009)] and structured sparsity [Baraniuk et al. (2010)] have been traditionally used.

Phase retrieval is another inverse imaging problem in several Fourier imaging applications, which involves reconstructing images from magnitude-only measurements. Compressive phase retrieval (CPR) models use sparse priors for reducing sample requirements; however, standard techniques from recent literature [Jagatap and Hegde (2017)] suggest a quadratic dependence of number of measurements on the sparsity level for recovering sparse images from magnitude-only Gaussian measurements and the design of a smart initialization scheme [Wang et al. (2017); Jagatap and Hegde (2017)]. If a prior is learned via a GAN [Hand et al. (2018)], [Shamshad et al. (2019)], then this requirement can be brought down; however one requires sufficient training data, which can be prohibitively expensive to obtain in domains such as medical or astronomical imaging.

1.2 Our contributions

In this paper, we explore, in depth, the use of untrained deep neural networks as an image prior for inverting images from under-sampled linear and non-linear measurements. Specifically, we assume that the image, \( x^{d \times 1} \) has \( d \) pixels. We further assume that the image \( x^* \) belongs to the range spanned by the weights of a deep under-parameterized untrained neural network \( G(\mathbf{w}; z) \),
which we denote by $\mathcal{S}$, where $\mathbf{w}$ is a set of the weights of the deep network and $\mathbf{z}$ is the latent code. The compressive measurements are stored in vector $\mathbf{y} = f(x^*)$, where $f$ embeds either compressive linear (defined by operator $A(\cdot)$) or compressive magnitude-only (defined by operator $|A(\cdot)|$) measurements. The task is to reconstruct image $\hat{x}$ which corresponds to small measurement error $\min_{x \in \mathcal{S}} \| f(x) - y \|_2^2$. With this setup, we establish theoretical guarantees for successful image reconstruction from both measurement schemes under untrained network priors.

Our specific contributions are as follows:

• We first present a new variant of the Restricted Isometry Property (RIP) Donoho (2006) via a covering number argument for the range of images $\mathcal{S}$ spanned by a deep untrained neural network. We use this result to guarantee unique image reconstruction for two different compressive imaging schemes.

• We propose a projected gradient descent (PGD) algorithm for solving the problem of compressive sensing with a deep untrained network prior. To our knowledge this is the first paper to use deep neural network priors for compressive sensing\footnote{We note recent concurrent work in [Heckel (2019)] which explores a similar approach for compressive sensing; however our paper focuses theoretical guarantees rooted in an algorithmic procedure.}, which relies on no training data\footnote{[Van Veen et al. (2018)] requires training data for learning a regularization function.}. We analyze the conditions under which PGD provably converges and report the sample complexity requirements corresponding to it. We also show superior performance of this framework via empirical results.

• We are the first to use deep network priors in the context of phase retrieval. We introduce a novel formulation, to solve compressive phase retrieval with fewer measurements as compared to state-of-art. We further provide preliminary guarantees for the convergence of a projected gradient descent scheme to solve the problem of compressive phase retrieval. We empirically show significant improvements in image reconstruction quality as compared to prior works.

We note that our sample complexity results rely on the number of parameters of the assumed deep network prior. Therefore, to get meaningful bounds, our network priors are under-parameterized, in that the total number of unknown parameters of the deep network is smaller than the dimension of the image. To ensure this, we build upon the formulation of the deep decoder [Heckel and...
Hand (2018)], which is a special network architecture resembling the decoder of an autoencoder (or generator of a GAN). The requirement of under-parameterization of deep network priors is natural; the goal is to design priors that concisely represent natural images. Moreover, this also ensures that the network does not fit noise [Heckel and Hand (2018)]. Due to these merits, we use select the deep decoder architecture for all analyses in this paper.

1.3 Prior work

Sparsifying transforms have long been used to constrain the solutions of inverse imaging problems in the context of denoising or inpainting. Conventional approaches to solve these problems include Basis Pursuit Denoising (BPDN) or Lasso [Chen et al. (2001)], TVAL3 [Li et al. (2009)], which rely on using $\ell_0$, $\ell_1$ and total variation (TV) regularizations on the image to be recovered. Sparsity based priors are highly effective and dataset independent, however it heavily relies on choosing a good sparsifying basis [Mallat (1999)].

Instead of hand-picking the sparsifying transform, in dictionary learning one learns both the sparsifying transform and the sparse code [Aharon et al. (2006)]. The dictionary captures global statistics of a given dataset. Multi-layer convolutional sparse coding [Sulam et al. (2018)] is an extension of sparse coding which models a given dataset in the form of a product of several linear dictionaries, all of which are convolutional in nature and this problem is challenging.

Generative adversarial networks (GAN) [Goodfellow et al. (2014)] have been used to generate photo-realistic images in an unsupervised fashion. The generator consists of stacked convolutions and maps random low-dimensional noise vectors to full sized images. GAN priors have been successfully used for inverse imaging problems [Bora et al. (2017); Hand et al. (2018); Hyder et al. (2019); Shah and Hegde (2018); Y. Wu (2019)]. The shortcomings of this approach are two-fold: test images are strictly restricted to the range of a trained generator, and the requirement of sufficient training data.

Sparse signal recovery from linear compressive measurements [Donoho (2006)] as well as magnitude-only compressive measurements [Jagatap and Hegde (2017)] has been extensively studied,

\footnote{Local structural information from a single image can also be used to learn dictionaries, by constructing several overlapping crops or patches of a single image.}
with several algorithmic approaches [Needell and Tropp (2009); Jagatap and Hegde (2017)]. In all of these approaches, modeling the low-dimensional embedding is challenging and may not be captured correctly using simple hand-crafted priors such as structured sparsity [Baraniuk et al. (2010)]. Since it is hard to estimate these hyper-parameters accurately, the number of samples required to reconstruct the image is often much higher than information theoretic limits [Jagatap and Hegde (2019); Bora et al. (2017)].

The problem of compressive phase retrieval specifically, is even more challenging because it is non-convex. Several papers in recent literature [Chen and Candes (2015); Cai et al. (2016); Jagatap and Hegde (2017)] rely on the design of a spectral initialization scheme which ensures that one can subsequently optimize over a convex ball of the problem. However this initialization requirement results in high sample requirements and is a bottleneck in achieving information theoretically optimal sample complexity.

Deep image prior [Ulyanov et al. (2018)] (DIP) uses primarily an encoder-decoder as a prior on the image, alongside an early stopping condition, for inverse imaging problems such as denoising, super-resolution and inpainting. Deep decoder [Heckel and Hand (2018)] (DD) improves upon DIP, providing a much simpler, underparameterized architecture, to learn a low-dimensional manifold (latent code) and a decoding operation from this latent code to the full image. Because it is under parameterized, deep decoder does not fit noise, and therefore does not require early stopping.

Deep network priors in the context of compressive imaging have only recently been explored [Van Veen et al. (2018)], and only in the context of compressive sensing. In contrast with [Van Veen et al. (2018)] which extends the idea of a Deep Image Prior to incorporate learned regularizations, in this paper we focus more on theoretical aspects of the problem and also explore applications in compressive phase retrieval. To our knowledge the application of deep network priors to compressive phase retrieval is novel.
Chapter 2. NOTATION

Throughout the paper, lower case letters denote vectors, such as \( v \) and upper case letters for matrices, such as \( M \). A set of variables subscripted with different indices is represented with bold-faced shorthand of the following form: \( \mathbf{w} := \{W_1, W_2, \ldots, W_L\} \). The neural network consists of \( L \) layers, each layer denoted as \( W_l \), with \( l \in \{1, \ldots, L\} \) and are \( 1 \times 1 \) convolutional. Up-sampling operators are denoted by \( U_l \). Vectorization of a matrix is written as \( \text{vec}(\cdot) \). The activation function considered is Rectified Linear Unit (ReLU), denoted as \( \sigma(\cdot) \). Hadamard or element-wise product is denoted by \( \circ \). Element-wise absolute valued vector is denoted by \( |v| \). Unless mentioned otherwise, \( \|v\| \) denotes vector \( \ell_2 \)-norm and \( \|M\| \) denotes spectral norm \( \|M\|_2 \).
Chapter 3. PROBLEM SETUP

3.1 Deep network priors

In this paper we discuss the problem of inverting a mapping \( x \to y \) of the form:

\[
y = f(x)
\]

where \( x = \text{vec}(X)^d_k \) is a \( d \)-dimensional signal \( X^{d \times k} \) (vectorized image), with \( k \) channels and \( f : x \to y \in \mathbb{R}^n \) captures a compressive measurement procedure, such as a linear operator \( A(\cdot) \) or magnitude only measurements \( |A(\cdot)| \) and \( n < dk \). We elaborate further on the exact structure of \( f \) in the next subsection (Section 3.2). The task of reconstructing image \( x \) from measurements \( y \) can be formulated as an optimization problem of the form:

\[
\min_{x \in S} \| y - f(x) \|_2^2 \tag{3.1}
\]

where we have chosen the \( \ell_2 \)-squared loss function and where \( S \) captures the prior on the image.

If the image \( x \) can be represented as the action of a deep generative network \( G(w; z) \) with weights \( w \) on some latent code \( z \), such that \( x = G(w; z) \), then the set \( S \) captures the characteristics of \( G(w; z) \). The latent code \( z := \text{vec}(Z_1) \) with \( Z_1 \in \mathbb{R}^{d_1 \times k_1} \) is a low-dimensional embedding with dimension \( d_1 k_1 \ll dk \) and its elements are generated from uniform random distribution.

When the network \( G(\cdot) \) and its weights \( w := \{W_1, \ldots W_L\} \) are known (from pre-training a generative network over large datasets) and fixed, the task is to obtain an estimate \( \hat{x} = G(w; \hat{z}) \), which indirectly translates to finding the optimal latent space encoding \( \hat{z} \). This problem has been studied in [Bora et al. (2017); Hand et al. (2018)] in the form of using learned GAN priors for inverse imaging.

In this paper however, the weights of the generator \( w \) are not pre-trained; rather, the task is to estimate image \( \hat{x} = G(w; z) \approx G(w^*; z) = x^* \) and corresponding weights \( \hat{w} \), for a fixed seed \( z \), where \( x^* \) is assumed to be the true image and the true weights \( w^* \) (possibly non-unique) satisfy
\[ w^* = \min_w \| x^* - G(z; w) \|_2^2. \] Note that the optimization in Eq. 3.1 is equivalent to substituting the surjective mapping \( G : w \to x \), and optimizing over \( w \),

\[
\min_w \| y - f(G(w; z)) \|_2^2, \tag{3.2}
\]
and estimate weights \( \hat{w} \) and corresponding image \( \hat{x} \).

Specifically, the untrained network \( G(w; z) \) takes the form of an expansive neural network; a decoder architecture similar to the one in [Heckel and Hand (2018)]\(^1\). The neural network is composed of \( L \) weight layers \( W_l \), indexed by \( l \in \{1, \ldots, L\} \) and are \( 1 \times 1 \) convolutions, upsampling operators \( U_l \) for \( l \in \{1, \ldots, L-1\} \) and ReLU activation \( \sigma(\cdot) \) and is expressed as follows

\[
x = G(w, z) = U_{L-1} \sigma(Z_{L-1} W_{L-1}) W_L = Z_L W_L, \tag{3.3}
\]
where \( \sigma(\cdot) \) represents the action of ReLU operation, \( Z_i^{d_i \times k_i} = U_{i-1} \sigma(Z_{i-1} W_{i-1}) \), for \( i = 2, \ldots, L \), \( z = \text{vec}(Z_1) \), \( d_L = d \) and \( W_L \in \mathbb{R}^{k_L \times k} \).

To capture the range of images spanned by the deep neural network architecture described above, we formally introduce the main assumption in our paper through Definition 1. Without loss in generality, we set \( k = 1 \) for the rest of this paper, while noting that the techniques carry over to general \( k \).

**Definition 1.** A given image \( x \in \mathbb{R}^d \) is said to obey an untrained neural network prior if it belongs to a set \( S \) defined as:

\[
S := \{ x | x = G(w, z) \}
\]
where \( z \) is a (randomly chosen, fixed) latent code vector and \( G(w; z) \) has the form in Eq. 3.3.

### 3.2 Observation models and assumptions

We now discuss the compressive measurement setup in more detail. Compressive measurement schemes were developed in [Donoho (2006)] for efficient imaging and storage of images and work

\(^1\)Alternatively, one may assume the architecture of the generator of a DCGAN [Radford et al. (2015); Van Veen et al. (2018)].
only as long as certain structural assumptions on the signal (or image) are met. The optimization problem in Eq. 3.1 is non-convex in general, partly dictated by the non-convexity of set $S$. Moreover, in the case of phase retrieval, the loss function is itself non-convex. Therefore unique signal recovery for either problems is not guaranteed without making specific assumptions on the measurement setup.

In this paper, we assume that the measurement operation can be represented by the action of a Gaussian matrix $A$ which is rank-deficient ($n < d$). The entries of this matrix are such that $A_{ij} \sim \mathcal{N}(0, 1/n)$. Linear compressive measurements take the form $y = Ax$ and magnitude-only measurements take the form $y = |Ax|$. We formally discuss the two different imaging schemes in the next two sections. We also present algorithms and theoretical guarantees for their convergence. For both algorithms, we require that a special $(S, \gamma, \beta)$-RIP holds for measurement matrix $A$, which is defined below.

**Definition 2.** $(S, \gamma, \beta)$-RIP: Set-Restricted Isometry Property with parameters $\gamma, \beta$:

For parameters $\gamma, \beta > 0$, a matrix $A \in \mathbb{R}^{n \times d}$ satisfies $(S, \gamma, \beta)$-RIP, if for all $x \in S$,

$$\gamma \|x\|^2 \leq \|Ax\|^2 \leq \beta \|x\|^2.$$  

We refer to the left (lower) inequality as $(S, \gamma)$-RIP and right (upper) inequality as $(S, \beta)$-RIP.

The $(S, 1 - \alpha, 1 + \alpha)$ RIP is achieved by Gaussian matrix $A$ under certain assumptions, which we state and prove via Lemma 1 as follows.

**Lemma 1.** If an image $x \in \mathbb{R}^d$ has a decoder prior (captured in set $S$), where the decoder consists of weights $w$ and piece-wise linear activation (ReLU), a random Gaussian matrix $A \in \mathbb{R}^{n \times d}$ with elements from $\mathcal{N}(0, 1/n)$, satisfies $(S, 1 - \alpha, 1 + \alpha)$-RIP, with probability $1 - e^{-\alpha^2 n}$, as long as $n = O \left( \frac{k_1}{\alpha^2} \sum_{l=2}^L k_l \log d \right)$, for small constant $c$ and $0 < \alpha < 1$.

**Proof sketch:** We use a union of sub-spaces model, similar to that developed in [Bora et al. (2017)] which was developed for GAN priors, to capture the range of a deep untrained network.

Our method uses a linearization principle. If the output sign of any ReLU activation $\sigma(\cdot)$ on its inputs were known a priori, then the mapping $x = G(w; z)$ becomes a product of linear weight
matrices and linear upsampling operators acting on the latent code $z$. The bulk of the proof relies on constructing a counting argument for the number of such linearized networks; call that number $N$. For a fixed linear subspace, the image $x$ has a representation of the form $x = UZw$, where $U$ absorbs all upsampling operations, $Z$ is latent code which is fixed and known and $w$ is the direct product of all weight matrices with $w \in \mathbb{R}^{k_1}$. An oblivious subspace embedding (OSE) of $x$ takes the form

$$(1 - \alpha)\|x\|^2 \leq \|Ax\|^2 \leq (1 + \alpha)\|x\|^2,$$

where $A$ is a Gaussian matrix, and holds for all $k_1$-dimensional vectors $w$, with high probability as long as $n = O(k_1/\alpha^2)$. We further require to take a union bound over all possible such linearized networks, which is given by $N$. The sample complexity corresponding to this bound is then computed to complete the set-restricted RIP result. The complete proof can be found in Section 8 and a discussion on the sample complexity is presented in Section 7.
Chapter 4. LINEAR COMPRESSIVE SENSING WITH DEEP UNTRAINED NETWORK PRIOR

We now analyze linear compressed Gaussian measurements of a vectorized image \( x \), with a deep network prior. The reconstruction problem assumes the following form:

\[
\min_x \| y - Ax \|^2 \quad \text{s.t.} \quad x = G(w; z),
\]

(4.1)

where \( A \in \mathbb{R}^{n \times d} \) is Gaussian matrix with \( n < d \), unknown weight matrices \( w \) and latent code \( z \) which is fixed. We solve this problem via Algorithm 1, Network Projected Gradient Descent (Net-PGD) for compressed sensing recovery.

Specifically, we break down the minimization into two parts; we first solve an unconstrained loss minimization of the objective function in Eq. 4.1 by implementing one step of gradient descent in Step 3 of Algorithm 1. The update \( v^t \) typically does not adhere to the deep network prior constraint \( v^t \not\in S \). To ensure that this happens, we solve a projection step in Line 4 of Algorithm 1, which happens to be the same as fitting a deep network prior to a noisy image. We iterate through this procedure in an alternating fashion until the estimates \( x^t \) converge to \( x^* \) within error factor \( \epsilon \).

We further establish convergence guarantees for Algorithm 1 in Theorem 1.

**Theorem 1.** Suppose the sampling matrix \( A \) satisfies \( (S, 1 - \alpha, 1 + \alpha) \)-RIP with high probability then, Algorithm 1 produces \( \hat{x} \) such that \( \| \hat{x} - x^* \| \leq \epsilon \), with \( \eta \) small enough, and requires \( T \propto \log \frac{1}{\epsilon} \) iterations.

**Proof sketch:** The proof of this theorem predominantly relies on our new set-restricted RIP result and uses standard techniques from compressed sensing theory. Indicating the loss function in Eq. 4.1 as \( L(x^t) = \| y - Ax^t \|^2 \), we aim to establish a contraction of the form \( L(x^{t+1}) < \nu L(x^t) \), with \( \nu < 1 \). To achieve this, we combine the projection criterion in Step 4 of Algorithm 1, which strictly implies that

\[
\| x^{t+1} - v^t \| \leq \| x^* - v^t \|
\]
Algorithm 1: Net-PGD for compressed sensing recovery.

1: **Input:** $y, A, z = \text{vec}(Z_1), \eta, T = \log \frac{1}{\epsilon}$
2: **for** $t = 1, \ldots, T$ **do**
3: \hspace{1em} $v^t \leftarrow x^t - \eta A^\top (Ax^t - y)$ \{gradient step for least squares\}
4: \hspace{1em} $w^t \leftarrow \arg \min_w \|v^t - G(w; z)\|$ \{projection to range of deep network\}
5: \hspace{1em} $x^{t+1} \leftarrow G(w^t; z)$
6: **end for**
7: **Output** $\hat{x} \leftarrow x^T$.

and $v^t = x^t - \eta A^\top (Ax^t - y)$ from Step 3 of Algorithm 1, where $\eta$ is chosen appropriately. Therefore,

$$\|x^{t+1} - x^t + \eta A^\top A(x^t - x^*)\|^2 \leq \|x^* - x^t + \eta A^\top A(x^t - x^*)\|^2.$$  

Furthermore, we utilize $(S, 1 - \alpha, 1 + \alpha)$-RIP and its Corollary 1 (refer Appendix A) which apply to $x^*, x^t, x^{t+1} \in S$, to show that

$$L(x^{t+1}) \leq \nu L(x^t)$$

and subsequently the error contraction $\|x^{t+1} - x^*\| \leq \nu_0\|x^t - x^*\|$, with $\nu, \nu_0 < 1$ to guarantee linear convergence of Net-PGD for compressed sensing recovery. This convergence result implies that Net-PGD requires $T \propto \log 1/\epsilon$ iterations to produce $\hat{x}$ within $\epsilon$-accuracy of $x^*$. The complete proof of Theorem 1 can be found in Section 8. In Section 5.1 we provide some exposition on the projection step (line 4 of Algorithm 1).
Chapter 5. COMPRESSIVE PHASE RETRIEVAL WITH DEEP UNTRAINED NETWORK PRIOR

In compressive phase retrieval, one wants to reconstruct a signal \( x \approx x^* \in S \) from measurements of the form \( y = |Ax^*| \) and therefore the objective is to minimize the following

\[
\min_x \| y - |Ax| \|^2 \quad \text{s.t.} \quad x = G(w; z),
\]  

(5.1)

where \( n < d \) and \( A \) is Gaussian, \( z \) is a fixed seed and weights \( w \) need to be estimated. We propose a Network Projected Gradient Descent (Net-PGD) for compressive phase retrieval to solve this problem, which is presented in Algorithm 2.

Algorithm 2 broadly consists of two parts. For the first part, in Line 3 we estimate the phase of the current estimate and in Line 4 we use this to compute the Wirtinger gradient [Chen and Candès (2015)] and execute one step for solving an unconstrained phase retrieval problem with gradient descent. The second part of the algorithm is (Line 5), estimating the weights of the deep network prior with noisy input \( v^t \). This is the projection step and ensures that the output \( w^t \) and subsequently the image estimate \( x^t = G(w^t; z) \) lies in the range of the decoder \( G(\cdot) \) outlined by set \( S \).

We highlight that the problem in Eq. 5.1 is significantly more challenging than the one in Eq. 4.1. The difficulty hinges on estimating the missing phase information accurately. For a real-valued vectors, there are \( 2^n \) different phase vectors \( p = \text{sign}(Ax) \) for a fixed choice of \( x \), which satisfy \( y = |Ax| \), moreover the entries of \( p \) are restricted to \( \{1, -1\} \). Hence, phase estimation is a non-convex problem. Therefore, with Algorithm 2 the problem in Eq.5.1 can only be solved to convergence locally; an initialization scheme is required to establish global convergence guarantees. We highlight the guarantees of Algorithm 2 in Theorem 2.

**Theorem 2.** Suppose the sampling matrix \( A^{n \times d} \) with Gaussian entries satisfies \( (S, 1 - \alpha, 1 + \alpha)\)-RIP with high probability, gradient descent solves Eq. 5.1 via Algorithm 2, such that \( \| \hat{x} - x^* \| \leq \epsilon \), as
Algorithm 2 Net-PGD for compressive phase retrieval.

1: **Input:** \( A, z = \text{vec}(Z_1), \eta, T = \log \frac{1}{\epsilon}, x^0 \) s.t. \( \| x^0 - x^* \| \leq \delta, \| x^* \| \).

2: **for** \( t = 1, \cdots, T \) **do**

3: \( p^t \leftarrow \text{sign}(Ax^t) \) \{phase estimation\}

4: \( v^t \leftarrow x^t - \eta A^\top (Ax^t - y \circ p^t) \) \{gradient step for phase retrieval\}

5: \( w^t \leftarrow \arg \min_w \| v^t - G(w; z) \| \) \{projection to range of deep network\}

6: \( x^{t+1} \leftarrow G(w^t; z) \)

7: **end for**

8: **Output** \( \hat{x} \leftarrow x^T \).

---

long as the weights are initialized appropriately, \( \eta \) is small enough, and the number of measurements
is \( n = O \left( k_1 \sum_{l=2}^L k_l \log d \right) \).

**Proof sketch:** The proof for Theorem 2 relies on two important results; \((S, 1 - \alpha, 1 + \alpha)\)-RIP and Lemma 2 which establishes a bound on the phase estimation error. Formally, the update in
Step 4 of Algorithm 2 can be re-written as

\[
v^{t+1} = x^t - \eta A^\top \left( Ax^t - Ax^* \circ \text{sign}(Ax^*) \circ \text{sign}(Ax^t) \right) = x^t - \eta A^\top \left( Ax^t - Ax^* \right) - \epsilon^t_p
\]

where \( \epsilon^t_p := A^\top Ax^* \circ (1 - \text{sign}(Ax^*) \circ \text{sign}(Ax^t)) \) is phase estimation error.

If \( \text{sign}(Ax^*) \approx \text{sign}(Ax^t) \), then the above resembles the gradient step from the linear compressive
sensing formulation. Thus, if \( x^0 \) is initialized well, the error due to phase mis-match \( \epsilon^t_p \) can be
bounded, and subsequently, a convergence result can be formulated.

Next, Step 4 of Algorithm 2 learns weights \( w^t \) that produce \( x^t = G(w^t; z) \), such that

\[
\| x^{t+1} - v^t \| \leq \| x^t - v^t \|
\]

for \( t = \{1, 2, \ldots T\} \). Then, the above projection rule yields:

\[
\| x^{t+1} - v^{t+1} + v^{t+1} - x^* \| \leq \| x^{t+1} - v^{t+1} \| + \| x^* - v^{t+1} \| \leq 2 \| x^* - v^{t+1} \|
\]

Using the update rule from Eq. 8.5 and plugging in for \( v^{t+1} \):

\[
\frac{1}{2} \| x^{t+1} - x^* \| \leq \| (1 - \eta A^\top A) h^t \| + \| \epsilon^t_p \|
\]
where $\eta$ is chosen appropriately. The rest of the proof relies on bounding the first term via matrix norm inequalities using Corollary 2 (in Appendix A) of $(S, 1 - \alpha, 1 + \alpha)$-RIP as $\|(1 - \eta A^T A)h^t\| \leq \rho_0\|h^t\|$ and the second term is bounded via Lemma 2 in Appendix A as $\|\varepsilon^p_t\| \leq \delta_0\|x^t - x^*\|$ as long as $\|x^0 - x^*\| \leq \delta_i\|x^*\|$. Hence we obtain a convergence criterion of the form

$$\|x^{t+1} - x^*\| \leq 2(\rho_0 + \eta\delta_0)\|x^t - x^*\| := \rho\|x^t - x^*\|.$$

where $\rho < 1$. Note that this proof relies on a bound on the phase error $\|\varepsilon^p_t\|$ which is established via Lemma 2. The complete proof for Theorem 2 can be found in Section 8. In Section 5.1 we provide some exposition on the projection step (line 5 of Algorithm 2). In our experiments (Section 6) we note that a uniform random initialization of the weights $w^0$ (which is common in training neural networks), to yield $x^0 = G(w^0; z)$ is sufficient for Net-PGD to succeed for compressive phase retrieval.
5.1 PROJECTION TO DEEP UNTRAINED NETWORK PRIOR

The projection steps in both Algorithms 1 and 2 represent the problem of fitting an image to an untrained neural network representation. This is the original setting for denoising and compression applications in [Ulyanov et al. (2018)] and [Heckel and Hand (2018)]. The algorithmic approach to solving this problem is via standard solvers such as gradient descent (GD) or Adam. The problem takes the form:

$$\min_w \mathcal{L}(w; z, v) := \min_w \|v - G(w; z)\|^2,$$

where $v$ is typically a noisy variant of the original image $x^*$. The problem in Eq.5.2 is non-convex due to the structure of $G(w; z)$. Convergence guarantees for deep neural network formulations of this form that exist are highly restrictive [Oymak and Soltanolkotabi (2019); Du et al. (2018)]. There exist several papers in recent literature which allude to (linear) convergence of gradient descent for solving the two-layer neural networks; however all of the results rely on moderate or extreme overparameterization of the neural network. Therefore, these results do not apply to our paper and deriving convergence guarantees for the denoising problem in 5.2 is an interesting direction for future work.
All experiments were run on a Nvidia GeForce GPU with 8GB RAM.

Dataset: We use images from the MNIST database and CelebA database to test our algorithms and reconstruct 6 grayscale (MNIST, 28 × 28 pixels (d = 784)) and 5 RGB (CelebA) images. The CelebA dataset images are center cropped to size 64 × 64 × 3 (d = 12288). The pixel values of all images are scaled to lie between 0 and 1.

Deep network architecture: We first optimize the deep network architecture which fit our example images such that \( x^* \approx G(w^*; z) \) (referred as “compressed” image). For MNIST images, the architecture was fixed to a 2 layer configuration \( k_1 = 15, k_2 = 15, k_3 = 10 \), and for CelebA images, a 3 layer configuration with \( k_1 = 120, k_2 = 15, k_3 = 15, k_4 = 10 \) was sufficient to represent most images. Both architectures use bilinear upsampling operators each with upsampling factor of \( 2, U_l^{i2}, l = \{1, 2, 3\} \). The outputs after each ReLU operation are normalized, by calling for batch normalization subroutine in Pytorch. Finally a sigmoid activation is added to the output of the deep network, which smoothens the output; however this is not mandatory for the deep network configuration to work. Note that both of these architectures are underparameterized, unlike the configurations in [Ulyanov et al. (2018)]. The random seed \( Z_1 \) is fixed and picked from uniform random distribution \(^1\). We plot the “compressed” representations of each image, \( G(w; z) \) in all Figures for reference.

Measurement setup: We use a Gaussian measurement matrix of size \( n \times d \) with \( n \) varied such that (i) \( n/d = 0.08, 0.1, 0.15, 0.2, 0.25, 0.3 \) for compressive sensing and (ii) \( n/d = 0.1, 0.2, 0.3, 0.5, 1, 3 \) for compressive phase retrieval. The elements of \( A \) are picked such that \( A_{i,j} \sim \mathcal{N}(0, 1/n) \) and we report averaged reconstruction error values over 10 different instantiations of \( A \) for a fixed image (image of digit ‘0’ from MNIST), network configuration and compression ratio \( n/d \).

\(^1\)Gaussian distributed entries as well as randomly picked rows of Hadamard matrices also work.
Figure 6.1 (CS) Reconstructed images from linear measurements (at compression rate \(n/d = 0.1\)) with (a) \(n = 78\) measurements for examples from MNIST, (b) \(n = 1228\) measurements for examples from CelebA, and (c) nMSE at different compression rates \(f = n/d\) for MNIST.

6.1 Compressive sensing

Algorithms and baselines: We implement 4 schemes based on untrained priors for solving CS, (i) gradient descent with deep network prior which solves Eq.3.2 (we call this Net-GD), similar to [Van Veen et al. (2018)] but without learned regularization (ii) Net-PGD, (iii) Lasso (\(\ell_1\) regularization) with sparse prior in DCT basis and finally (iv) TVAL3 [Li et al. (2009)] (Total Variation regularization). The TVAL3 code only works for grayscale images, therefore we do not use it for CelebA examples. The reconstructions are shown in Figure 6.1 for images from (a) MNIST and (b) CelebA datasets.

Implementation details: For CS recovery with deep network priors, both Net-GD and Net-PGD were implemented using the PyTorch framework with Python 3 and using GPU support. For Net-GD, SGD (alternatively, Adam) optimizer is used. For Net-PGD, SGD (alternatively, Adam) optimizer is used for the projection step and SGD optimizer for the gradient step in Step 3 of Alg. 1 and Step 4 of Alg. 2. For implementing Lasso algorithm, Python’s `sklearn.linear_model` library was used and we set the regularization factor \(\alpha = 10^{-5}\). The MATLAB code for TVAL3 [Li et al. (2009)] made available on the author’s website was used with its default settings.
Figure 6.2 (CPR) Reconstructed images from magnitude-only measurements (a) at compression rate of $n/d = 0.3$ for MNIST, (b) at compression rates of $n/d = 0.1, 0.5$ for CelebA with (row 1,3) Net-GD and (row 2,4) Net-PGD, (c) nMSE at different compression rates $f = n/d$ for MNIST.

Performance metrics: We compare reconstruction quality using normalized Mean-Squared Error (nMSE), which is calculated as $\|\hat{x} - x^*\|^2/\|x^*\|^2$. We plot the variation of the nMSE with different compression rates $f = n/d$ for all the algorithms tested averaged over all trials for MNIST in Figure 6.1 (c). We note that both Net-GD and Net-PGD produce superior reconstructions as compared to state of art.

Running time: We also report the average running times for different algorithms across different measurement levels for examples from MNIST is 5.86s (Net-GD), 5.46s (Net-PGD), 2.43s (Lasso-DCT), 0.82s (TVAL3). We note that the running time of both GD and PGD for CS-UNP are competitive.

6.2 Compressive phase retrieval

Algorithms and baselines: We implement 3 schemes based on untrained priors for solving CPR, (i) Net-GD (ii) Net-PGD and finally (iii) Sparse Truncated Amplitude Flow (Sparta) [Wang et al. (2017)], with sparse prior in DCT basis for both datasets. The reconstructions are shown in Figure 6.2 for (a) MNIST and (b) CelebA datasets. We plot nMSE at varying compression rates for all
algorithms averaged over all trials for MNIST in Figure 6.2(c) and note that both Net-GD and Net-PGD outperform Sparta.

Implementation details: For compressive phase retrieval with deep network priors, both Net-GD and Net-PGD were implemented using the PyTorch framework with Python 3 and using GPU support. All optimization procedures were implemented using SGD optimizer. For implementing Sparta algorithm, the algorithm from [Wang et al. (2017)] was implemented in MATLAB.

Running time: We report the average running times for different algorithms across different measurement levels for examples from MNIST is 25.59s (Net-GD), 28.46s (Net-PGD), 3.80s (Sparta-DCT).

Goodness of random initialization: Our theoretical guarantees for phase retrieval hold only as long as the initialization $x^0$ is close to the ground truth $x^*$. We perform rigorous experiments to assert that uniform random initialization of the weights $w^0$ of the neural network, ensure that the initial estimate $x^0 = G(w^0; z)$ is good. We denote the distance of initialization as $\delta_i = \|x^0 - x^T\|/\|x^T\|$ ($x^T = \hat{x}$) and report the values of $\delta_i$ for the trials in which $\|x^T - x^*\|/\|x^*\| < 0.1$. We plot the average values of $\delta_i$ in Table 6.1.

<table>
<thead>
<tr>
<th>n/d</th>
<th>d channel configuration</th>
<th>nMSE of $\hat{x}$</th>
<th>average $\delta_i$ values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>784 (MNIST) 15, 15, 10</td>
<td>0.098</td>
<td>0.914</td>
</tr>
<tr>
<td>0.5</td>
<td>784 (MNIST) 15, 15, 10</td>
<td>0.018</td>
<td>0.942</td>
</tr>
<tr>
<td>0.4</td>
<td>12288 (CelebA) 120, 15, 15,10</td>
<td>0.020</td>
<td>0.913</td>
</tr>
<tr>
<td>0.6</td>
<td>12288 (CelebA) 120, 15, 15,10</td>
<td>0.015</td>
<td>0.915</td>
</tr>
</tbody>
</table>

From our observation, uniform random initialization suffices to ensure that the conditions for Theorem 2 are met and $\delta_i < 1$. 
Chapter 7. DISCUSSION ON SAMPLE COMPLEXITY

In compressive imaging literature, for $s$-sparse signals of dimension $d$, the sample complexity for compressive sensing is $n = O(s \log d)$ and compressive phase retrieval is $n = O(s^2 \log d)$, when Gaussian measurements are considered. If structural constraints are imposed on the sparsity of images, such as block sparsity, the sample requirements can be brought down to $n = O(s/b \log d)$ and $n = O(s^2/b \log d)$ for CS and CPR respectively, where $b$ is the block length of each sparse block [Jagatap and Hegde (2017)]. However these gains come at the cost of designing the signal priors carefully.

In contrast, the sample requirements with deep network priors, as we show in this paper is $n = O(k_1 \sum_{l=2}^{L} k_l \log d)$. In both datasets that we tested, relatively shallow architectures were sufficient. Therefore the effective sample complexity is of the order of $k_1$, which is typically much smaller than the dimension $d$. We have empirically demonstrated in Section 6 that the sample requirement with deep network priors is significantly lower than that for the sparse prior setting. Moreover, the design of the prior is fairly straightforward, and applies for a wide class of images.
Chapter 8. PROOFS OF MAIN LEMMAS AND THEOREMS

In this section we proofs for the main theorems and lemmas discussed in the main body of this paper.

We first discuss the set-restricted restricted isometry property.

The \((S, \gamma, \beta)\) RIP holds for Gaussian matrix \(A\) with high probability, as long as certain dimensionality requirements are met. We show this via Lemma 1 as follows:

**Lemma 1.** If an image \(x \in \mathbb{R}^d\) has a decoder prior (captured in set \(S\), where the decoder consists of weights \(w\) and piece-wise linear activation (ReLU), a random Gaussian matrix \(A \in \mathbb{R}^{n \times d}\) with elements from \(N(0, 1/n)\), satisfies \((S, 1 - \alpha, 1 + \alpha)\)-RIP, with probability \(1 - e^{-ca^2n}\), as long as \(n = O\left(\frac{k_1^2 \sum_{l=2}^{L} k_l \log d}{\alpha^2}\right)\), for small constant \(c\) and \(0 < \alpha < 1\).

**Proof.** We first describe the two layer setup.

Consider the action of measurement matrix \(A\) defined on vector \(h\), where \(h := U_1 \sigma(ZW_1)W_2\) below:

\[
\begin{align*}
  u &= Ah = AU_1 \sigma(ZW_1)W_2.
\end{align*}
\]

where \(W_1^{k_1 \times k_2}\), \(W_1^{k_2 \times 1}\) and \(U_1^{d \times d_1}\) with \(d > d_1\).

We would like to estimate the dimensionality of \(A\), required to ensure that the action of \(A\) on set restricted vector \(h \in S\), is bounded as:

\[
\gamma \|h\|^2 \leq \|Ah\|^2 \leq \beta \|h\|^2
\]

with high probability. To establish this, consider the following argument which is similar to the union of subspaces argument from [Bora et al. (2017)].

The action of ReLU on input \((ZW_1)\) partitions the input space of variable \(W_1\) into a union of linear subspaces. In particular, consider a single column of \(w_{1,j}\) of \(W_1\), indexed by \(j\), which is \(k_1\) dimensional. Then, \(\sigma(Zw_{1,j})\) partitions the \(k_1\)-dimensional input space into \((d_1^{k_1})\) \(k_1\)-spaces.
(Lemma 3 in Appendix A). Since there are $k_2$ such columns, effectively the $k_1 \times k_2$ dimensional space of $W_1$ is partitioned into $(d_1^{k_1})^k$, $(k_1 \times k_2)$-spaces.

Then, we can consider the union of $d_1^{k_1k_2}$ subspaces with linearized mappings of the form:

$$u_1 = AU_1(Z_1W'_1)W_2$$

where $W_1'$ belongs to one of the $d_1^{k_1k_2}$ subspaces and $u_1$ is the mapping corresponding to that.

If the dimensionalities are chosen such that they satisfy $d > k_2$, and $A, U_1, Z_1$ are known matrix operators, then the effectively $w^{k_1 \times 1} := W'_1W_2$ represents the accumulated action of the weights, belonging to one of the $d_1^{k_1k_2}$ subspaces, $(U_1Z_1)^d \times k_1$ is a linear transformation from a lower dimensional space to a higher dimensional space. Then, if $A$ is designed as an oblivious subspace embedding (OSE) (Lemma 4 in Appendix A) of $U_1Z_1w$, for a single $k_1$-dimensional subspace of $w$, one requires $m = O\left(\frac{k_1}{\alpha^2}\right)$ samples to embed the vector $w$, as

$$(1 - \alpha)\|h\|^2 \leq \|Ah\|^2 \leq (1 + \alpha)\|h\|^2,$$  

with probability $1 - e^{-\alpha_1^2n}$, for constant $\alpha_1 < \alpha$. Since there are $d_1^{k_1k_2}$ such subspaces, then for the OSE to hold for all subspaces, one requires to take a union bound as $1 - d_1^{k_1k_2}e^{-\alpha_1^2n}$. Therefore the expression in Eq. 8.1 holds for all $h \in S$, with probability $1 - e^{-\alpha_2^2n}$ and $\alpha_2 < \alpha_1$. Therefore, one requires $n = O\left(\frac{k_1k_2 \log d_1}{\alpha_1^2}\right)$, to ensure that $A$ satisfies $(S, 1 - \alpha, 1 + \alpha)$-RIP with probability $1 - e^{-\alpha_2^2n}$.

**Multiple layers:** A similar argument can be extended for multiple layers. Consider an $L$ layer formulation:

$$u = AU_{L-1}\sigma(\ldots\sigma(U_1\sigma(Z_1W_1)W_2\ldots)W_3\ldots)W_L$$

with $W_{L}^{k_L \times 1}$ and $U_{L-1}^{d \times d_{L-1}}$.

The first non-linearity partitions the space into $d_1^{k_1k_2}$ $k_1 \times k_2$-dimensional spaces. Thus we have the part-linearized mapping of the form:

$$u_1 = AU_{L-1}\sigma(\ldots U_2\sigma(U_1Z_1W'_1W_2)W_3\ldots)W_L$$

and there are $d_1^{k_1k_2}$ of these.
The second non-linearity acts on input \((U_1Z_1)^{d_2 \times k_1} \cdot (W_1'W_2)^{k_1 \times k_3}\) of each of these partitions, and creates more partitions; \(d_2^{k_1k_3}\) partitions of the \(k_1 \times k_3\) space. This creates effectively \(d_1^{k_1k_2} \times d_2^{k_1k_3} \leq d_2^{k_1(k_2+k_3)}\) (since \(d_2 > d_1\)) partitions in total and these constitute linearized embeddings of the form:

\[
U = AU_{L-1} \sigma(\ldots \sigma(U_2U_1Z_1W_1'W_2')W_3\ldots)W_L
\]

where \(W_1'W_2'\) belong to one of the \(d_1^{k_1k_2} \cdot d_2^{k_1k_3}\) subspaces.

Extending the same argument to all subsequent non-linearities (total \((L-1)\) such) and linearizing, we have mappings of the form

\[
U_{L-1} = AU_{L-1}(\ldots (U_2U_1Z_1W_1'W_2')W_3')W_L
\]

\[
h_{L-1} = \left( \prod_{l=1}^{L-1} U_l \right) Z_1 \cdot \left( \prod_{l=1}^{L} W_l \right)
\]

\[
\quad = B \cdot w
\]

where \(B := \left( \prod_{l=1}^{L-1} U_l \right) Z_1\) and \(w := \left( \prod_{l=1}^{L} W_l \right) \in \mathbb{R}^{k_1}\). The total number of partitions are \(d_1^{k_1k_2} \times d_2^{k_1k_3} \ldots d_1^{k_1} \sum_{l=2}^{L} k_l\), since \(d > d_{L-1} > \ldots d_1\), via upsampling operations. Effectively we consider a union of \(d_1^{k_1} \sum_{l=2}^{L} k_l\) subspaces of dimension \(k_1\).

Repeating the argument from the analysis for two layers, if \(A\) is designed as an oblivious subspace embedding (OSE) (Lemma 4 in Appendix A) of \(B \cdot w\), for a single \(k_1\)-dimensional subspace of \(Bw\), one requires \(m = O\left( k_1^2 \right)\) samples to embed the vector \(w\), with the bound in Eq. 8.1 with probability \(1 - e^{-\alpha_1^2 n}\), for constant \(\alpha_1 < \alpha\).

Therefore, the embedding from Eq. 8.1 holds for

\[
h = U_{L-1} \sigma(\ldots \sigma(U_1 \sigma(ZW_1)W_2)W_3\ldots)W_L,
\]

as long as \(n = O\left( \frac{k_1 \sum_{l=2}^{L} k_l \log d}{\alpha_1^2} \right)\), with probability \(1 - e^{-\alpha_1^2 n}\), which implies that \(A\) satisfies \((\mathcal{S}, 1-\alpha, 1+\alpha)\)-RIP with high probability.

\[\square\]

In Appendix A, we present some corollaries (Corollary 1 and 2) of the above result which will be useful for proving some of our theoretical claims.
Next, we discuss the convergence of Net-PGD for compressed sensing recovery via Theorem 1.

**Theorem 1.** Suppose the sampling matrix $A^{n \times d}$ satisfies $(S, 1 - \alpha, 1 + \alpha)$-RIP with high probability then, Algorithm 1 produces $\hat{x}$ such that $\|\hat{x} - x^*\| \leq \epsilon$, with $\eta$ small enough, and requires $T \propto \log \frac{1}{\epsilon}$ iterations.

**Proof.** Using the definition of loss as $L(x^t) = \|y - Ax^t\|^2$,  

$$L(x^{t+1}) - L(x^t) = (\|Ax^{t+1}\|^2 - \|Ax^t\|^2) - 2(y^\top Ax^{t+1} - y^\top Ax^t)$$  

$$= \|Ax^{t+1} - Ax^t\|^2 - 2(Ax^t)^\top (Ax^t) + 2(Ax^t)^\top (Ax^{t+1})$$  

$$- 2(y^\top Ax^{t+1} - y^\top Ax^t)$$  

$$= \|Ax^{t+1} - Ax^t\|^2 - 2(y - Ax^t)^\top (Ax^{t+1} - Ax^t) \tag{8.3}$$

We want to establish a contraction of the form $L(x^{t+1}) < \nu L(x^t)$, with $\nu < 1$.

Step 3 of Algorithm 1 is solved via gradient descent:

$$v^t = x^t - \eta A^\top (Ax^t - Ax^*) \tag{8.4}$$

Subsequently, Step 4 of Algorithm 1 learns weights $w^t$ that produce $x^t = G(w^t; z)$, which lies in the range of the decoder $G(\cdot)$ and is closest to the estimate $v^t$.

Step 4 of Algorithm 1 produces an update of $w^t$ satisfying:

$$\|G(w^t; z) - v^t\| \leq \|G(w^*; z) - v^t\|$$

Denoting $G(w^t; z) := x^t$ and $G(w^*; z) := x^*$, and using the update rule in Eq. 8.4,  

$$\|x^{t+1} - v^t\|^2 \leq \|x^* - v^t\|^2$$  

$$\|x^{t+1} - x^t + \eta A^\top A(x^t - x^*)\|^2 \leq \|x^* - x^t + \eta A^\top A(x^t - x^*)\|^2$$  

$$\|x^{t+1} - x^t\|^2 + 2\eta(A(x^t - x^*))^\top A(x^{t+1} - x^*) \leq \|x^t - x^*\|^2 - 2\eta\|A(x^t - x^*)\|^2$$  

$$\frac{1}{\eta}\|x^{t+1} - x^t\|^2 + 2(A(x^t - x^*))^\top A(x^{t+1} - x^*) \leq \frac{1}{\eta}\|x^t - x^*\|^2 - 2L(x^t)$$  

$$\implies L(x^{t+1}) + L(x^t) \leq \frac{1}{\eta}\|x^t - x^*\|^2 - \frac{1}{\eta}\|x^{t+1} - x^t\|^2$$  

$$+ \|A(x^{t+1} - x^t)\|^2$$
where we have used the expansion in Eq. 8.3. We now use \((S, \gamma, \beta)\)-RIP. If a Gaussian measurement matrix is considered then \(\gamma = 1 - \alpha\) and \(\beta = 1 + \alpha\).

Using \((S, \gamma)\)-RIP on the first term on the right side,
\[
\|x^* - x^t\|^2 \leq \frac{1}{\gamma} \|A(x^* - x^t)\|^2
\]
Second, using \((S, \beta)\)-RIP on the last term on the right side,
\[
\|A(x^{t+1} - x^t)\|^2 \leq \beta \|x^{t+1} - x^t\|^2
\]
Accumulating these expressions and substituting,
\[
L(x^{t+1}) + L(x^t) \leq \frac{1}{\eta \gamma} L(x^t) + \left(\beta - \frac{1}{\eta}\right) \|x^{t+1} - x^t\|^2
\]
\[
\leq \frac{\beta \eta < 1}{\eta \gamma^2} L(x^t)
\]
\[
\implies L(x^{t+1}) \leq \nu L(x^t)
\]
\[
\implies L(x^T) \leq \nu^T L(x^0)
\]
where \(0 < \nu < 1\) and \(\nu = \left(\frac{1}{\eta \gamma^2} - 1\right)\) and picking \(\eta < 1/\beta\). Invoking \((S, \gamma, \beta)\)-RIP again,
\[
\|x^T - x^*\|^2 \leq \frac{1}{\gamma} \|y - Ax^T\|^2 \leq \frac{\nu^T}{\gamma} \|y - A x^0\|^2 := \epsilon
\]
Hence to reach \(\epsilon\)-accuracy in reconstruction, one requires \(T\) iterations where
\[
T = \log_{\nu} \left(\frac{\|y - A x^0\|^2}{\gamma \epsilon}\right).
\]
Note that the contraction \(L(x^{t+1}) \leq \nu L(x^t)\) coupled with \((S, \gamma, \beta)\)-RIP implies a distance contraction
\[
\|x^{t+1} - x^*\| \leq \nu_o \|x^t - x^*\|, \text{ with } \nu_o = \nu \sqrt{\beta/\gamma}.
\]
Step 4 of Algorithm 1, which is essentially the case of fitting a noisy image to a deep neural network prior can be solved via gradient descent. We discuss this projection in further detail in Section 5.1.

Next, we discuss the main convergence result of Net-PGD for compressive phase retrieval in Theorem 2.
Theorem 2. Suppose the sampling matrix $A^{n \times d}$ with Gaussian entries satisfies $(S, 1 - \alpha, 1 + \alpha)$-RIP with high probability, gradient descent solves Eq. 5.1 via Algorithm 2, such that $\|\hat{x} - x^*\| \leq \epsilon$, as long as the weights are initialized appropriately, $\eta$ is small enough, and the number of measurements is $n = O \left( k_1 \sum_{l=2}^L k_l \log d \right)$.

Proof. Step 4 of Algorithm 2 is solved via a variant of gradient descent called Wirtinger flow Zhang and Liang (2016a), which produces updates of the form:

$$\begin{align*}
v^{t+1} &= x^t - \eta A^\top (Ax^t - Ax^* \circ \text{sign}(Ax^*) \circ \text{sign}(Ax^t)) \\
      &= x^t - \eta A^\top (Ax^t - Ax^*) - \eta A^\top Ax^* \circ (1 - \text{sign}(Ax^*) \circ \text{sign}(Ax^t)) \\
      &= x^t - \eta A^\top (Ax^t - Ax^*) - \eta \epsilon_p^t \\
\end{align*}$$

(8.5)

where $\epsilon_p^t := A^\top Ax^* \circ (1 - \text{sign}(Ax^*) \circ \text{sign}(Ax^t))$ is phase estimation error.

If $\text{sign}(Ax^*) \approx \text{sign}(Ax^t)$, then the above resembles the gradient step from the linear compressed sensing formulation. Thus, if $x^0$ is initialized well, the error due to phase mis-match $\epsilon_p^t$ can be bounded, and subsequently, a convergence result can be formulated.

Next, Step 5 of Algorithm 2 learns weights $w^t$ that produce $x^t = G(w^t; z)$, which lies in the range of the decoder $G(\cdot)$ and is closest to the estimate $v^t$. We discuss this projection in further detail in Appendix 5.1.

Step 5 of Algorithm 2 produces an update of $w^t$ satisfying:

$$
\|G(w^t; z) - v^t\| \leq \|G(w^*; z) - v^t\| \equiv \|x^t - v^t\| \leq \|x^* - v^t\|
$$

for $t = \{1, 2, \ldots, T\}$. Then, the above projection rule yields:

$$
\|x^{t+1} - v^{t+1} + v^{t+1} - x^*\| \leq \|x^{t+1} - v^{t+1}\| + \|x^* - v^{t+1}\| \leq 2\|x^* - v^{t+1}\|
$$

Using the update rule from Eq. 8.5 and plugging in for $v^{t+1}$:

$$
\frac{1}{2}\|x^{t+1} - x^*\|^2 \leq \|(x^t - x^*) - (\eta A^\top (Ax^t - Ax^*) + \eta \epsilon_p^t)\|^2
$$
Defining $h^{t+1} = x^{t+1} - x^*$ and $h^t = x^t - x^*$, the above expression is
\[
\frac{1}{2} \|h^{t+1}\| \leq \|h^t - \eta A^T A h^t - \eta \varepsilon^t_p\| \leq \|(1 - \eta A^T A)h^t\| + \eta \|\varepsilon^t_p\|
\]  \hspace{1cm} (8.6)

We now bound the two terms in the expression above separately as follows. The first term is bounded using matrix norm inequalities Using Corollary 2 (in Appendix A) of $(S, \gamma, \beta)$-RIP:
\[
\|(1 - \eta A^T A)h^t\| \leq \max\{1 - \eta \lambda_{\min}, \eta \lambda_{\max} - 1\} \|h^t\|
\]
where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues of $A^T A$ restricted on set $S$, and via Corollary 2, $\lambda_{\min} = (1 - \alpha)$, $\lambda_{\max} = (1 + \alpha)$.

Hence the first term in the right side of Eq.8.6 is bounded as:
\[
\|(1 - \eta A^T A)h^t\| \leq \rho_o \|h^t\|
\]
where $\rho_o = \max\{1 - \eta(1 - \alpha), \eta(1 + \alpha) - 1\}$. The second term in Eq.8.6 is bounded via Lemma 2 as follows:
\[
\|\varepsilon^t_p\| \leq \delta_o \|x^t - x^*\|
\]
as long as $\|x^0 - x^*\| \leq \delta \|x^*\|$.

Substituting back in Eq.8.6,
\[
\|x^{t+1} - x^*\| \leq 2(\rho_o + \eta \delta_o) \|x^t - x^*\| := \rho \|x^t - x^*\|.
\]
Then, if we pick constant $\eta = \frac{1}{1 + \alpha + 1 - \alpha} = 1$ that minimizes $\rho := 2(\max\{1 - \eta(1 - \alpha), \eta(1 + \alpha) - 1\} + \eta \delta_o)$, to yield $\rho = 2(\alpha + \delta_o)$ then we obtain the linear convergence criterion as follows:
\[
\|x^{t+1} - x^*\| \leq \rho \|x^t - x^*\|.
\]
Here, if we set $\alpha = 0.1$ and $\delta_o = 0.36$ from Lemma 2, then $\rho = 0.92 < 1$. Note that this proof relies on a bound on the phase error $\|\varepsilon^t_p\|$ which is established via Lemma 2 as follows:

**Lemma 2.** Given initialization condition $\|x^0 - x^*\| \leq \delta \|x^*\|$, then if one has Gaussian measurements $A \in \mathbb{R}^{n \times d}$ such that $n = O\left(k_1 \sum_{i=2}^L k_l \log d\right)$, then with probability $1 - e^{-c_2 n}$, the following holds:
\[
\|\varepsilon^t_p\| = \|A^T A x^* \circ \left(1 - \text{sign}(A x^* \circ \text{sign}(A x^t))\right)\| \leq \delta_o \|x^t - x^*\|
\]
for constant $c_2$ and $\delta_o = 0.36$. 

Proof of Lemma 2 can be found in Appendix A.


A. SUPPLEMENTARY LEMMAS

Corollary 1. For parameter $\alpha > 0$, if a matrix $A \in \mathbb{R}^{n \times d}$ satisfies $(S, 1 - \alpha, 1 + \alpha)$-RIP with probability $1 - e^{-c_2n}$, for all $x \in S$, then for $x_1, x_2 \in S$,

$$(1 - \alpha)\|x_1 - x_2\|^2 \leq \|A(x_1 - x_2)\|^2 \leq (1 + \alpha)\|x_1 - x_2\|^2,$$

holds with probability $1 - e^{-c_2n}$, where $c_2 < c$.

Proof. Since $x_1, x_2 \in S$, both $x_1, x_2$ lie in the union of $k_1$-dimensional subspaces, the difference vector $x_3 = x_1 - x_2 \in S'$, lies in a union of $2k_1$-dimensional subspaces. For $(S, 1 - \alpha, 1 + \alpha)$-RIP to hold for the difference set, one continues to require $n = O\left(\frac{k_1 \sum_{l=2}^{k_1} \log d}{\alpha_1^2}\right)$. \(\square\)

Corollary 2. If $A$ satisfies set-restricted RIP and $h^t = x^t - x^*$, with $x^t, x^* \in S$ then

$$\|(1 - \eta A^\top A)h^t\| \leq \max \{1 - \eta \lambda_{\min}, \eta \lambda_{\max} - 1\}\|h^t\|$$

with $\lambda_{\min} = (1 - \alpha)$ and $\lambda_{\max} = (1 + \alpha)$.

Proof. Consider $h \in S'$, where $h = h^t = x^t - x_2$ and $x^t, x^* \in S$. Then from Set-RIP and Corollary 1,

$$(1 - \alpha)\|h\|^2 \leq \|Ah\|^2 \leq (1 + \alpha)\|h\|^2.$$

From Eq. 8.2, if $x_1, x_2 \in S$, then it is possible to write $h$ to arise from a union of $2k_1$-dimensional subspaces of the form $h = Bw$. Then,

$$(1 - \alpha)\|Bw\|^2 \leq \|ABw\|^2 \leq (1 + \alpha)\|Bw\|^2$$

(A.1)

where $w \in \mathbb{R}^{2k_1}$. We need to evaluate the eigenvalues of $\|A^\top A\|$ restricted on set $S'$, which we can do by inducing a projection on the union of subspaces $B$ as

$$\|A^\top Ah\| = \|B^\top A^\top ABw\|$$
Therefore, the minimum and maximum eigenvalues of $\|A^\top A\|$ restricted on set $S'$ are

$$\sigma_{\text{min}}(AB) \leq \|B^\top A^\top A\|_2 \leq \sigma_{\text{max}}(AB)$$

Then, using Eq.A.1, $(1 - \alpha)\sigma_{\text{min}}(B) \leq \|B^\top A^\top A\|_2 \leq (1 + \alpha)\sigma_{\text{max}}(B)$.

Since $B$ predominantly consists of a product of upsampling matrices and latent code $Z_1$, which can be always chosen such that $\sigma_{\text{max}}(Z_1) \approx \sigma_{\text{min}}(Z_1)$, therefore $\sigma_{\text{max}}(B) \approx \sigma_{\text{min}}(B) \approx 1$.

**Lemma 2.** Given initialization condition $\|x^0 - x^*\| \leq \delta_i \|x^*\|$, then if one has Gaussian measurements $A \in \mathbb{R}^{n \times d}$ such that $n = O \left( k_1 \sum_{l=2}^{L} k_l \log d \right)$, then with probability $1 - e^{-c_2 n}$, the following holds:

$$\|\epsilon_p^t\| = \|A^\top Ax^* \circ (1 - \text{sign}(Ax^*) \circ \text{sign}(Ax^t))\| \leq \delta_o \|x^t - x^*\|$$

for constant $c_2$ and $\delta_o = 0.36$.

**Proof.** We adapt the proof of Lemma C.1. of Jagatap and Hegde (2019) as follows.

We define indicator function $1_{(a_i^\top x^t)(a_i^\top x^*)<1} = \frac{1}{2}(1 - \text{sign}(Ax^*) \circ \text{sign}(Ax^t))$ with zeros where the condition is false and ones where the condition is true.

Then we are required to bound the following expression:

$$\|\epsilon_p^t\|^2 = 2 \sum_{i=1}^{n} (a_i^\top x^*)^2 \cdot 1_{(a_i^\top x^t)(a_i^\top x^*)<1} \leq \delta_o^2 \|x^t - x^*\|^2$$

Following the sequence of arguments in Lemma C.1. of Jagatap and Hegde (2019) (or Lemma C.1 of Zhang and Liang (2016b)), one can show that for a given $x^t$,

$$\|\epsilon_p^t\|^2 \leq \delta_o^2 + \kappa + \frac{3c_1 \kappa}{\delta_i} < 0.13 + \kappa + \frac{3c_1 \kappa}{\delta}$$

(A.2)

with high probability, $1 - e^{-cn \kappa^2}$, for small constants $c, c_1, \delta$, as long as $\|x^t - x^*\| \leq 0.1 \|x^*\|_2$.

Here the bound on $\delta_o^2$ (in this case 0.13) is a monotonically increasing function of the distance $\delta_i = \frac{\|x^t - x^*\|_2}{\|x^*\|_2}$.

If the projected gradient scheme produces iterates satisfying

$$\|x^{t+1} - x^*\| < \rho \|x^t - x^*\|$$
with $\rho < 1$, then the condition in Eq. A.2 is satisfied for all $t = \{1, 2, \ldots, T\}$ as long as the initialization $x^0$ satisfies $\|x^0 - x^*\| \leq 0.1\|x^*\|_2$ (i.e. $\delta^0_i := \delta_i = 0.1$).

Now, the expression in Eq. A.2 holds for a fixed $x^t$. To ensure that it holds for all possible $x \in S$, we need to use an epsilon-net argument over the space of variables spanned by $S$. The cardinality of $S$ is

$$\text{card}(S) < d^{k_1 \sum_{l=2}^{L} k_l}$$

as seen from the derivation of RIP in Lemma 1. Therefore,

$$\|e^t\| \leq 0.13 + \kappa + \frac{3c_1 \kappa}{\delta_i}$$

with probability $1 - d^{k_1 \sum_{l=2}^{L} k_l} e^{-c n \kappa^2}$ for small constant $c$. To ensure that high probability result holds for all $x \in S$,

$$e^{k_1 \sum_{l=2}^{L} k_l \log d} e^{-c n \kappa^2} < e^{-c_2 n}$$

$$k_1 \sum_{l=2}^{L} k_l \log d - c n \kappa^2 < -c_2 n$$

$$n > \frac{1}{c \kappa^2 - c_2} k_1 \sum_{l=2}^{L} k_l \log d > c_3 k_1 \sum_{l=2}^{L} k_l \log d$$

for appropriately chosen constants $c, c_2, c_3$.

Note that this Theorem requires that the weights are initialized appropriately, satisfying $\|x^0 - x^*\| \leq \delta_i \|x^*\|$. In Section 6 we perform rigorous experiments to show that random initialization suffices to ensure that $\delta_i$ is small.

We state the partitioning argument, which is borrowed from Lemma 8.3 of Bora et al. (2017) directly, as follows.

**Lemma 3.** Number of partitions Bora et al. (2017). Consider $d$ different $k - 1$ dimensional hyperplanes in $\mathbb{R}^k$. Then, the number of $k$-dimensional partitions (also called $k$-faces) such that relative to each partitioning hyperplane, all points inside a partition are on the same side, is $O(d^k)$.

Finally we state the statement for Oblivious Subspace Embedding, which is the core theoretical lemma required for proving our RIP result.
Lemma 4. Oblivious subspace embedding (OSE) S. (2006). A $(k, \alpha, \delta)$-OSE is a random matrix $\Pi^{n \times d}$ such that for any fixed $k$-dimensional subspace $S$ and $x^{d \times 1} \in S$, with probability $1 - \delta$, $\Pi$ is a subspace embedding for $S$ with distortion $\alpha$, where $n = O(\alpha^{-2}(k + \log(\frac{1}{\delta}))$.

The failure probability is $\delta = e^{-cn\alpha^2 + ck}$, for small constant $c$ and the embedding satisfies:

$$(1 - \alpha)\|x\|^2 \leq \|\Pi x\|^2 \leq (1 + \alpha)\|x\|^2.$$